Marcos Araújo Castro e Silva

Doutorando em Genética e Biologia Evolutiva no IB/USP, trabalha com análise de ancestralidade e miscigenação em populações humanas brasileiras.

Contato: marcosaraujocastro@gmail.com

Meus Exercícios

Exercícios

Proposta de Trabalho Final

Plano A: Equilíbrio de Hardy-Weiberg

Contextualização: o que é o Equilíbrio de Hardy-Weinberg?

O Equilíbrio de Hardy-Weinberg (EHW) é um modelo nulo para estudos evolutivos e em genética de populações, em outras palavras ele descreve uma série de pressupostos que quando verdadeiros determinam a não ocorrência de alteração das frequências gênicas (alélicas) ao longo das gerações. Os pressupostos do modelo de Hardy-Weinberg são?

A) Organismos diplóides;

B) Reprodução exclusivamente sexuada;

C) Não há sobreposição de gerações;

D) Frequências alélicas idênticas entre os sexos;

E) Acasalamentos ao acaso;

F) População infinita (muito grande);

G) Não há migração,

H) Não há mutação;

I) Seleção natural ignorável.

Estas são as premissas clássicas do modelo, porém adotaremos aqui mais duas, por hora ao menos,

para evitar tornar a função demasiadamente complexa, que é a ocorrência de apenas 2 alelos por gene e que estes genes sejam autossomais (este pressoposto é comumente adotado). Quando nenhum dos pressupostos do modelo é quebrado, este permite a previsão das frequências genotípicas e alélicas da próxima geração apenas com o conhecimento das frequências gênicas da atual, pois as frequências se mantem constantes.

O que a função vai fazer?

1) Testar se a(s) população(ões) encontram-se em EHW, através dos seguintes passos:

. Calcular as frequências alélicas a partir de frequências genotípicas ou valores de contagem de indivíduos, para uma ou múltiplas populações;

. Estimar as frequências genotípicas esperadas sobre EHW;

. Realizar o teste de x^2 de aderência entre os valores observados e esperados, de modo a determinar se a(s) população(ões) encontra-se sob EHW para o gene de interesse;

. Determinar o p-valor associado a estatística x^2 calculada, inferir se a(s) população(ões) encontra(m)-se ou não sobre EHW, de acordo com o valor de significância escolhido pelo usuário.

2) A função poderá ainda inferir sob a existência de estrutura populacional e endogamia. Por meio do cálculo do índice de fixação, que é basicamente a probabilidade de sortear um indivíduo com 2 alelos idênticos por descendência, podendo ainda particionar a probabilidade devida a endogamia e aquela associada a deriva genética (e portanto estrutura populacional). Estes índices estão relacionado à redução do número de indivíduos heterozigotos observados em relação aos esperados, dentro de cada subpopulação (FIS), da heterozigosidade esperada dentro de cada subpopulação em relação a esperada para a população como um todo (FST) e do número de heterozigotos observados e esperados na população como um todo (FIT). Os passos são:

. Calcular o FIS (a probabilidade de sortear um indivíduo com 2 alelos idênticos por descendência devido a endogamia);

. Calcular o FST (a probabilidade de sortear um indivíduo com 2 alelos idênticos por descendência devido a deriva genética);

. Calcular o FIT (a probabilidade total de sortear um indivíduo com 2 alelos idênticos por descendência).

Função e argumentos

Função: genpop(dados, modo= “ewh”, sig=0.05)

1º Argumento (dados): um objeto da classe “dataframe” com genótipos nas colunas (3 colunas: 2 para homozigotos/ 1 para heterozigotos) e com as populações nas linhas (quantas forem necessárias).

2º Argumento (modo): uma “string” de caracteres específica que condiciona o modo de

funcionamento da função:

. modo = “ehw”: determina que a função irá testar se a(s) população(ões) encontram-se em EHW;

. modo = “fit”: irá calcular o índice de fixação para as populações, as porções devidas à deriva genética e à endogamia.

3º Argumento (sig): um valor númerico indicando a significância dos testes estatísticos a serem realizados. Não precisa existe para o modo “fit”.

Resultados e objetos de saída

- modo “ehw”: Produz um objeto da classe “list” contendo nas duas primeiras posições, 2 vetores contendo caracteres identificadores das populações, no primeiro das populações que não desviam do EHW e no segundo as populações que desviam.

No restante das posições são colocados n objetos da classe “dataframe”, onde n é o número de populações estudadas. Cada objeto contendo 4 colunas: número de indivíduos observados, número de indivíduos esperados, valor do teste x^2 calculado e p-valor. E nas linhas os 3 genótipos possíveis homozigotos e heterozigoto.

- modo “fit”: Produz um objeto da classe “dataframe” contendo 3 colunas com os nomes FIS, FST e FIT e os valores numéricos das estimativas dos índices.

Possíveis incrementos e desafios

- Lidar com problemas como por exemplo dados faltantes e frequências que somam mais do que 1 (ou 100%).

- Possibilitar os cálculos para genes com 3 alelos e para genes ligados ao sexo (preciso pensar mais em como operacionalizar).

- Incluir outros índices de interesse em genética de populações, como outros modos de funcionamento da função (argumento “modo”).

Comentários Lucas

Hey, Marcos!
Gostei bastante dessa função. Pela forma com que você apresentou sua proposta, me pareceu que você tem bastante segurança sobre o assunto e, então, a parte desafiadora fica a cargo do uso da linguagem. Esta proposta me parece útil e de certa forma, mais trabalhosinha, então incentivo você a ficar com ela. Uma coisa que fiquei pensando seria a necessidade *real e oficial* dessa lista como objeto de retorno no caso do modo “ehw”. Eu entendi o porque dela, mas fiquei pensando se poderíamos substituir ela por algum objeto dataframe que seja mais fácil de trabalhar (mas só se tiver).
Também, sendo bastante honesto com você, eu entendo pouco de genética de populações e por isso pode ser que eu esteja falando

algo sem sentido, mas gostei muito da ideia de gráfico que você apresentou na proposta B (eu gosto de gráficos!) e fiquei me perguntando se não seria interessante algum plot que permita ao usuário entender a distribuição dessas frequências alélicas entre as populações. O que você acha? Teria alguma fundamentação teórica para isso?
Qualquer coisa, estamos aí! Abraços,
Lucas

Comentários Marcos

Oi Lucas, tudo bom?

Fico satisfeito que tenha gostado da proposta da função, também concordo que a primeira proposta será um desafio maior e também que gerará um produto mais útil depois. Quanto ao output da função, eu acho que já encontrei a solução, que seria mais ou menos a seguinte: eu colocaria todos os dados em um único dataframe, com uma população seguida pelas outras nas linhas subsequentes, conforme a tabela abaixo. E nas últimas duas colunas, primeiro aparece o valor do cálculo do p-valor, com base no x^2 acumulado para as três classes (os 3 genótipos) e depois uma coluna com valores lógicos, indicando se aquela população se encontra ou não em EHW. No exemplo o x^2 acumulado é dado por 0.002 + 0.05 + 0.3 = 0.352 e o p-valor é calculado a partir desse resultado pela função "pchisq()" do R, que para o exemplo é 0.4470168. Neste exemplo, se adotarmos um nível de significância (probabilidade do erro tipo 1) para o teste de 0.05, não se rejeita a hipótese de que a população em questão está sob EHW.

Como o p-valor é apenas um para cada população, ele aparece 3 vezes na tabela para cada população, assim como o valor lógico indicando se a população está ou não em EHW. Além disso a última coluna elimina a necessidade de produção das listas (em objetos da classe "vector") de populações que estão ou não em EHW. Portanto possibilita a função tenha como única saída um objeto da classe "dataframe".

Agradeço as suas dicas, já foram muito úteis. Por enquanto ainda estou pensando em como fazer cada um dos passos, mas já tenho esquematizado o desenho geral da função. A medida que surgirem dúvidas eu te pergunto.

Muito obrigado! Abraços,

População	**Genótipo**	**Observado**	**Esperado**	**x^2**	**p-valor**	**EHW**
1	AA	835	833.6	0.002	0.4470168	TRUE
1	Aa	156	158.9	0.05	0.4470168	TRUE
1	aa	9	7.5	0.3	0.4470168	TRUE
2	AA					
2	Aa					
2	aa					
3	AA					

População	Genótipo	Observado	Esperado	x²	p-valor	EHW
3	Aa					
3	aa					

Comentários Lucas

Hey, Marcos!
Desculpe a demora em responder. Em campo aqui a internet nem sempre está muito amigável... rs

Obrigado pelas explicações! Eu gostei da sua alternativa, acho que assim sua função vai ter um output mais “clean” e objetivo.
E sim!! Qualquer coisa, pode me escrever por email, se achar melhor, que eu consigo ver pelo celular mais rápido. Aí eu respondo :)

Abraços,
Lucas

Plano B: Simulação de deriva genética

O que a função vai fazer?

Realiza uma simulação de Deriva genética de “i” populações com tamanho populacional “n” para um gene bialélico ao longo de “x” gerações. Ao final produz uma tabela de resultados das simulações, um gráfico mostrando o que acontece ao longo do tempo e o resultado de um cálculo de probabilidade.

Função e argumentos

Função: simder(p, n, g=100, prob=F)

1º Argumento (p): Um valor numérico ou um objeto da classe “vector” contendo valores numéricos, os quais representam a frequência de um dos alelos, a frequência do outro alelo é obtida pelo complemento (q=1-p), pois como são genes bialélicos, a frequência dos 2 alelos deve somar 1. O número de valores contidos pelo vetor é igual ao número “i” de populações a serem simuladas.

2º Argumento (n): Um valor inteiro ou um objeto da classe “vector” contendo valores inteiros que configuram o tamanho populacional. Na mesma forma do primeiro argumento, o número de valores colocados no vetor se refere à quantidade de populações a serem simuladas. Portando os vetores colocados em p e n devem possuir o mesmo comprimento.

3º Argumento (g): Um valor inteiro determinando o número de gerações simuladas. Por padrão é feita uma simulação com 100 gerações.

4º Argumento (prob): Um valor lógico (TRUE ou FALSE), indicando se a função deve fazer o cálculo da probabilidade da frequência de um alelo se fixar em 0 e 1, baseado nas simulações da função. Por exemplo, se x é a quantidade de vezes em que um alelo com a frequência p inicial se fixa em 1, e n é o número de simulações executadas, a probabilidade de que um alelo com frequência p inicial é dada

por Probabilidade=x/n. Quando é fornecido o valor TRUE para esse argumento, todas as populações devem apresentar o mesmo tamanho populacional (“n”), para que seja possível o cálculo da probabilidade. Caso o valor fornecido seja FALSE a função não faz o cálculo da probabilidade. Por padrão o valor fornecido ao argumento é FALSE.

Resultados e objetos de saída

1) Um objeto da classe “Dataframe” contendo “i” colunas com as frequências alélicas de p, para cada uma das “i” populações e nas linhas figuram as “g” gerações.

2) Um gráfico de linhas, com as “g” gerações no eixo X e os valores de p no eixo Y. Cada uma das “i” populações é representada como uma linha no gráfico, com uma cor e tipo de linha diferente para cada.

3) Argumento prob=T: Uma mensagem informando a probabilidade da frequência alélica se fixar em 1 e 0.

Possíveis incrementos e desafios

- Incluir na função a simulação de outros processos e forças evolutivas.

Comentários Lucas

Hey, Marcos!
Também gostei bastante dessa função. Você a apresentou de forma clara e concisa. Contudo, a primeira proposta me pareceu mais desafiadora e útil, pois permite ao usuário usá-la para diferentes objetivos. Eu investiria na primeira proposta.

Proposta selecionada

Plano A: Equilíbrio de Hardy-Weiberg (com as modificações sugeridas):

Contextualização: o que é o Equilíbrio de Hardy-Weinberg?

O Equilíbrio de Hardy-Weinberg (EHW) é um modelo nulo para estudos evolutivos e em genética de populações, em outras palavras ele descreve uma série de pressupostos que quando verdadeiros determinam a não ocorrência de alteração das frequências gênicas (alélicas) ao longo das gerações. Os pressupostos do modelo de Hardy-Weinberg são?

A) Organismos diplóides;

B) Reprodução exclusivamente sexuada;

C) Não há sobreposição de gerações;

D) Frequências alélicas idênticas entre os sexos;

E) Acasalamentos ao acaso;

F) População infinita (muito grande);

G) Não há migração,

H) Não há mutação;

I) Seleção natural ignorável.

Estas são as premissas clássicas do modelo, porém adotaremos aqui mais duas, por hora ao menos, para evitar tornar a função demasiadamente complexa, que é a ocorrência de apenas 2 alelos por gene e que estes genes sejam autossomais (este pressoposto é comumente adotado). Quando nenhum dos pressupostos do modelo é quebrado, este permite a previsão das frequências genotípicas e alélicas da próxima geração apenas com o conhecimento das frequências gênicas da atual, pois as frequências se mantem constantes.

Neste sentido a automatização do teste de hipótese sobre o Equilíbrio de Hardy-Weinberg possibilita inferir rapidamente se existem ou não forças (ou processos) evolutivas atuando sobre grandes conjuntos de dados, contendo muitas populaçãoes. Este tipo de inferência somado a análise visual das frequências genotípicas observadas e esperadas sob EHW possui não apenas utilidade prática, mas pode também ser utilizado como instrumental didático no ensino de genética de populações e evolução.

O que a função vai fazer?

1) Testar se a(s) população(ões) encontram-se em EHW, através dos seguintes passos:

. Sumarizar o número de indivíduos portadores de cada genótipo e o tamanho populacional de cada população;

. Calcular as frequências genotípicas observadas;

. Calcular as frequências alélicas a partir de frequências genotípicas observadas;

. Estimar as frequências genotípicas esperadas sobre EHW;

. Realizar o teste de x^2 de aderência entre os valores observados e esperados, de modo a determinar se a(s) população(ões) encontra-se sob EHW para o gene de interesse;

. Determinar o p-valor associado a estatística x^2 calculada e inferir se a(s) população(ões) encontra(m)-se ou não sobre EHW, de acordo com um valor de significância definido pelo usuário.

2) A função poderá ainda inferir sobre a existência de estrutura populacional e endogamia. Por meio do cálculo do índice de fixação, que é basicamente a probabilidade de sortear um indivíduo com 2 alelos idênticos por descendência, podendo ainda particionar a probabilidade devido à endogamia e aquela associada à estrutura populacional (e portanto à deriva genética). Estes índices estão relacionado à redução do número de indivíduos heterozigotos observados em relação aos esperados,

dentro de cada subpopulação (FIS), da heterozigosidade esperada dentro de cada subpopulação em relação a esperada para a população como um todo (FST) e do número de heterozigotos observados e esperados na população como um todo (FIT). Os passos são:

. Calcular o FIS (a probabilidade de sortear um indivíduo com 2 alelos idênticos por descendência devido a endogamia);

. Calcular o FST (a probabilidade de sortear um indivíduo com 2 alelos idênticos por descendência devido a deriva genética);

. Calcular o FIT (a probabilidade total de sortear um indivíduo com 2 alelos idênticos por descendência).

Função e argumentos

Função: genpop(dados, modo="ehw", alfa=0.05, input="freq", graphics="on")

1º Argumento (dados): um objeto da classe "data.frame" contendo nas linhas as populações (quantas forem necessárias) e nas colunas as frequências (valor numérico entre 0 e 1) genotípicas observadas de um gene bialélico (3 colunas = 2 homozigotos + 1 heterozigoto), seguidas de uma coluna contendo o tamanho populacional de cada população (totalizando 4 colunas), neste caso deve-se utilizar o argumento input = "freq"; Ou um objeto da classe "data.frame" contendo nas linhas os indivíduos (quantos forem necessários) e nas colunas o genótipo do indivíduo e a população a que este pertence (totalizando 2 colunas).

2º Argumento (modo): uma "string" de caracteres específica que condiciona o modo de funcionamento da função; modo = "ehw" determina que a função irá testar se a(s) população(ões) encontram-se em Equilíbrio de Hardy-Weinberg; modo = "fit": determina que a função irá calcular o índice de fixação nas suas porções FIS, FST e FIT.

3º Argumento (alfa): um valor númerico indicando a significância do teste estatístico a ser realizado; É desconsiderado no modo "fit".

4º Argumento (input): uma "string" de caracteres que especifica o tipo de arquivo de entrada usado na função; input = "freq" quando o objeto de entrada é o padrão da função, com valores de frequências genotípicas e tamanho populacional para cada população; input = "raw" quando o objeto de entrada contem apenas a informação sobre o genótipo e a população de cada indivíduo.

5º Argumento (graphics): uma "string" de caracteres que determina se serão construídos gráficos das frequências genotípicas observadas e esperadas, assim como das frequências alélicas; graphics = "on" para que os gráficos sejam gerados; graphics = "off" para que os gráficos não sejam gerados.

Resultados e objetos de saída

- modo "ehw": uma tabela (classe: data.frame) com as populações nas linhas e nas colunas os valores de indivíduos observados e esperados sobre EHW para cada genótipo, bem como os valores de X^2 calculados, p-valor e o resultado de um teste lógico, que compara o p-valor estimado para cada população com o nível de significância do teste estipulado pelo usuário através do argumento "alfa".

- modo “fit”: uma tabela (classe: data.frame) contendo 3 colunas, uma para cada índice (FIT, FST e FIS) e apenas uma linha com os valores dos índices calculados.

Em ambos os modos, dado o argumento 'graphics' = “on”, são gerados 2 tipos de gráficos para cada população do conjunto de dados analisados. Um gráfico de barras contendo as frequências genotípicas observadas como barras em cinza e as frequências genotípicas esperadas sob Equilíbrio de Hardy-Weinberg como barras transparentes com bordas vermelhas sobrepostas as barras anteriores, por fim no eixo Y encontra-se uma escala de frequência e no X os diferentes genótipos. Também é gerado um segundo gráfico de barras com os valores de frequências alélicas, na forma de barras em cinza e com a escala de frequência no eixo Y e os alelos no eixo X.

Página de ajuda da função

```
genpop                 package: -                R Documentation

Realiza o Teste de X^2 de Aderência para determinar se uma ou mais
populações encontram-se em Equilíbrio de Hardy-Weinberg e alternativamente
calcula o Índice de Fixação (FIT, FST e FIS). Além disso, produz gráficos
das frequências genotípicas observadas e esperadas, bem como das frequências
alélicas.

Description:

Testa se diferentes populações encontram-se em Equilíbrio de Hardy-Weinberg
pelo teste de X^2 de Aderência, comparando o p-valor calculado a um nível de
significância do teste estipulado pelo usuário ou calcula o Índice de
Fixação, em seus componentes FIT, FST e FIS. A partir de uma tabela (classe:
data.frame) contendo frequências genotípicas observadas de um gene bialélico
e tamanho populacional para diferentes populações ou de uma tabela (classe:
data.frame) contendo o genótipo e a população a que cada indivíduo pertence.

Usage:

genpop(dados, modo="ehw", alfa=0.05, input="freq", graphics="on")

Arguments:

dados       um objeto da classe “data.frame” contendo nas linhas as
populações (quantas forem necessárias) e nas colunas as frequências (valor
numérico entre 0 e 1) genotípicas observadas de um gene bialélico (3 colunas
= 2 homozigotos + 1 heterozigoto), seguidas de uma coluna contendo o tamanho
populacional de cada população (totalizando 4 colunas), neste caso deve-se
utilizar o argumento input = "freq"; Ou um objeto da classe "data.frame"
contendo nas linhas os indivíduos (quantos forem necessários) e nas colunas
o genótipo do indivíduo e a população a que este pertence (totalizando 2
colunas). Ver a seção 'Examples' desta documentação da função para exemplos
de objetos de entrada.

modo        uma “string” de caracteres específica que condiciona o modo de
```

funcionamento da função; modo = “ehw” determina que a função irá testar se a(s) população(ões) encontram-se em Equilíbrio de Hardy-Weinberg; modo = “fit”: determina que a função irá calcular o índice de fixação nas suas porções FIS, FST e FIT.

alfa um valor númerico indicando a significância do teste estatístico a ser realizado; É desconsiderado no modo “fit”.

input uma “string” de caracteres que especifica o tipo de arquivo de entrada usado na função; input = "freq" quando o objeto de entrada é o padrão da função, com valores de frequências genotípicas e tamanho populacional para cada população; input = "raw" quando o objeto de entrada contem apenas a informação sobre o genótipo e a população de cada indivíduo.

graphics uma “string” de caracteres que determina se serão construídos gráficos das frequências genotípicas observadas e esperadas, assim como das frequências alélicas; graphics = "on" para que os gráficos sejam gerados; graphics = "off" para que os gráficos não sejam gerados.

Value:

Em seu uso padrão, argumento modo = "ehw", a função produz uma tabela (classe: data.frame) com as populações nas linhas e nas colunas os valores de indivíduos observados e esperados sobre EHW para cada genótipo, bem como os valores de X^2 calculados, p-valor e o resultado de um teste lógico, que compara o p-valor estimado para cada população com o nível de significância do teste estipulado pelo usuário através do argumento "alfa". Quando o p-valor ($P[X \leq x]$) é menor do que 1 - "alfa", o teste retorna o valor TRUE na coluna EHW, indicando que a população está em EHW.

A segunda funcionalidade, argumento modo = "fit", gera uma tabela (classe: data.frame) contendo 3 colunas, uma para cada índice (FIT, FST e FIS) e apenas uma linha com os valores dos índices calculados.

Em ambos os modos, dado o argumento 'graphics' = "on", são gerados 2 tipos de gráficos para cada população do conjunto de dados analisados. Um gráfico de barras contendo as frequências genotípicas observadas como barras em cinza e as frequências genotípicas esperadas sob Equilíbrio de Hardy-Weinberg como barras transparentes com bordas vermelhas sobrepostas as barras anteriores, por fim no eixo Y encontra-se uma escala de frequência e no X os diferentes genótipos. Também é gerado um segundo gráfico de barras com os valores de frequências alélicas, na forma de barras em cinza e com a escala de frequência no eixo Y e os alelos no eixo X.

Warning:

A função emite mensagens de erro nas seguintes situações: quando pelo menos um valor das frequências genotípicas observadas não está contido no intervalo de 0 a 1; se pelo menos um valor de tamanho populacional não é maior do que 0; se houver qualquer valor faltante (NA) no objeto inserido no

```
argumento 'dados'; se a string de caracteres fornecida para o argumento
'modo' é diferente de "ehw" e "fit"; quando o valor fornecido ao argumento
'alfa' não está contido no intervalo de 0 a 1; se o objeto inserido no
argumento 'dados' não é um data.frame; a função também retorna uma mensagem
(Warning message) quando o argumento graphics = "off", avisando que os
gráficos estão desativados.

Note:

Atentar para o fato de que os objetos de entrada devem ser objetos da classe
"data.frame", com a formatação descrita na seção 'Arguments' desta
documentação e evitando os erros mais comuns descritos na seção 'Warning'
desta documentação, para que a função opere corretamente. Por isso
recomenda-se realizar previamente uma análise exploratória dos dados de modo
a encontrar, corrigir ou omitir os erros e dados faltantes para que a função
possa ser utilizada.

Para tornar possível visualizar os gráficos gerados recomenda-se utilizar o
software RStudio (https://www.rstudio.com/), navegando através dos gráficos
pela aba 'Plots' da interface gráfica do mesmo. Caso contrário é necessário
alterar os parâmetros da construção de gráficos para permitir que todos os
gráficos sejam gerados sem sobreposição em um único dispositivo gráfico, por
exemplo através do argumento 'mfrow' da função 'par' (ver seção 'Examples'
desta documentação). Entretanto, para um número elevado de populações
recomenda-se particionar o conjunto de dados em porções menores de
populações para a entrada na função e criação dos gráficos. É interessante
também criar um dispositivo gráfico de vizualização através da função 'X11'
e maximizar a janela criada antes de plotar os gráficos, de modo que a
utilização do espaço seja otimizada e evite a sobreposição dos elementos
gráficos.

Author(s):

Marcos Araújo Castro e Silva
macsilva@usp.br

São Paulo, 23 de Junho de 2017.

References:

RIDLEY, M (2006). Evolution. Oxford: Blackwell Science Ltd. 752 p.

See Also:

Funcoes: X11(), par().

Examples:

## Input padrão:
pops=c("Esquimós","Aborígenes
Australianos","Egípcios","Alemães","Chineses","Nigerianos") # Cria um vetor
```

```
com o nome das populações.
MM=c(0.835,0.024,0.278,0.297,0.332,0.301) # Cria um vetor com os valores de
frequências do genótipo MM para cada população.
Mm=c(0.156,0.304,0.489,0.507,0.486,0.495) # Cria um vetor com os valores de
frequências do genótipo Mm para cada população.
mm=c(0.009,0.672,0.233,0.196,0.182,0.204) # Cria um vetor com os valores de
frequências do genótipo mm para cada população.
n=c(rep(1000,6)) # Cria um vetor contendo valores de tamanho populacional
para cada população, nesse caso todas as populações possuem 1000 indivíduos.
humans=data.frame(MM,Mm,mm,n) # Cria um objeto contendo o data frame com as
frequências genotípicas.
rownames(humans)=pops # Renomeia as linhas com os nomes das populações
contidos no objeto pops.
rm(pops, MM, Mm, mm, n) # Remove os objetos usados para criar o dataframe de
exemplo.

x11() # Cria um dispositivo gráfico de visualização. MAXIMIZE A JANELA ANTES
DE PLOTAR OS GRÁFICOS, assim você evita a sobreposição de elementos
gráficos, conforme discutido na seção 'Notes' da documentação.
par(mfrow=c(4,3)) # Altera os parâmetros gráficos, para permitir a criação
de 12 gráficos no dispositivo gráfico (4 linhas e 3 colunas).
genpop(humans) # Utiliza a função 'genpop' sobre o objeto 'humans', com os
argumentos padrões da função (genpop(dados, modo="ehw", alfa=0.05,
input="freq", graphics="on")).
par(mfrow=c(1,1)) # Retorna ao padrão os parâmetros gráficos.

genpop(humans, graphics = "off") # Utiliza a função com a criação de
gráficos desabilitada.

genpop(humans, alfa = 0.1) # Utiliza a função com o nível de significância
para o teste de X^2 (argumento alfa) igual a 0.1.

genpop(humans, modo = "fit") # Utiliza a função no modo de cálculo do Índice
de fixação.

## Input bruto:
Pop1 <- sample(c("AA","Aa","aa"), 1000, replace=TRUE, prob=c(0.3, 0.5, 0.2))
# Cria vetores contendo cada um dos genótipos (AA, Aa, aa), amostrados com
as probabilidades definidas pelo argumento prob.
Pop2 <- sample(c("AA","Aa","aa"), 500, replace=TRUE, prob=c(0.7, 0.15,
0.15))
Pop3 <- sample(c("AA","Aa","aa"), 650, replace=TRUE, prob=c(0.3, 0.45,
0.25))
Pop4 <- sample(c("AA","Aa","aa"), 2000, replace=TRUE, prob=c(0.2, 0.6, 0.2))
Pop5 <- sample(c("AA","Aa","aa"), 850, replace=TRUE, prob=c(0.05, 0.9,
0.05))
Pop6 <- sample(c("AA","Aa","aa"), 5000, replace=TRUE, prob=c(0.3, 0.6, 0.1))
alleles=c(Pop1,Pop2,Pop3,Pop4,Pop5,Pop6) # Cria um vetor a partir da
concatenação dos vetores criados anteriormente.
alleles=data.frame(gen=alleles, pop=rep(paste("Pop",c(1:6),sep=""),
```

```
times=c(1000,500,650,2000,850,5000))) # Cria um data.frame a partir do vetor 'alleles' e de um vetor contendo a população a que cada indivíduo pertence.
rm(Pop1,Pop2,Pop3,Pop4,Pop5,Pop6) # Remove os objetos usados para criar o dataframe de exemplo.


genpop(alleles, input = "raw") # Utiliza a função 'genpop' sobre o objeto 'alleles', com os argumentos padrões da função, exceto o argumento input = "raw", aqui utilizado pelo natureza do objeto de entrada.

genpop(alleles, modo = "fit", input = "raw") # Utiliza a função no modo de cálculo de Índice de fixação.
```

Código da função

```
genpop=function(dados, modo="ehw", alfa=0.05, input="freq", graphics="on"){
# Uso básico da função com os parâmetros definidos.
  ##### PADRONIZANDO INPUTS BRUTOS:
  ### Produzindo uma tabela (classe="data.frame") com dados de frequências genotípicas e tamanho populacional, no caso do argumento 'input' = "raw":
  if(input=="raw"){ # Controle de fluxo que determina a execução desta porção do código quando o argumento 'input' é igual a "raw". Transformando a informação dos genótipos de indivíduos e atribuição a populações, em um data.frame de entrada padrão da função.
    dados=as.data.frame(cbind(t(table(dados)), n=apply(t(table(dados)), 1, sum))) # Criando um data.frame com o número de indivíduos de cada genótipo nas 3 primeiras colunas e o tamanho populacional (n) na quarta e última coluna. Em cada linha é colocada uma população.
    dados[,c(1:3)]=dados[,c(1:3)]/dados[,c(4)] # Transformando os valores de número de indivíduos por genótipo em frequências genotípicas. O que é feito através da divisão do número de indivíduos de cada genótipo, pelo tamanho populacional de cada população.
  }
  ##### TESTANDO OS INPUTS:
  ### Testando as entradas da função por meio de controle de fluxo, para alguns problemas mais gerais e caso eles ocorram emitindo algumas mensagens de erro específicas.
  if(modo!="fit" & modo!="ehw"){ # Controle de fluxo que interrompe a execução da função se a string de caracteres fornecida ao argumento 'modo' não é uma das opções válidas (e. g. "ehw" e "fit").
    stop("Erro: Por favor forneça uma das opções válidas para o argumento 'modo': ehw ou fit") # Interrope a função e emite a mensagem de erro entre aspas, caso a condição anterior do controle de fluxo se cumpra.
  }
  if(0>alfa | 1<alfa){ # Controle de fluxo que interrompe a execução da função quando o valor fornecido ao argumento 'alfa' não está contido no intervalo de 0 a 1.
    stop("Erro: O valor fornecido ao argumento 'alfa' deve estar contido no intervalo de 0 a 1") # Interrope a função e emite a mensagem de erro entre
```

```
aspas, caso a condição anterior do controle de fluxo se cumpra.
  }
  if(!is.data.frame(dados)){ # Controle de fluxo que interrompe a execução
da função quando o objeto fornecido ao argumento 'dados' não é um
data.frame.
    stop("Erro: O objeto inserido no argumento 'dados' não é um dataframe")
# Interrope a função e emite a mensagem de erro entre aspas, caso a condição
anterior do controle de fluxo se cumpra.
  }
  if(sum(is.na(dados))>=1){ # Controle de fluxo que interrompe a execução da
função se o objeto fornecido ao argumento 'dados' contem algum dado faltante
(NA).
    stop("Erro: Existe pelo menos um dado faltante (NA) no objeto inserido
no argumento 'dados'") # Interrope a função e emite a mensagem de erro entre
aspas, caso a condição anterior do controle de fluxo se cumpra.
  }
  if(sum(dados[,c(1:3)]<0 | dados[,c(1:3)]>1)>=1){ # Controle de fluxo que
interrompe a execução da função se pelo menos um dos valores de frequências
genotípicas não está contido no intervalo de 0 a 1.
    stop("Erro: Existe pelo menos um valor de frequência genotípica que não
está contido no intervalo de 0 a 1") # Interrope a função e emite a mensagem
de erro entre aspas, caso a condição anterior do controle de fluxo se
cumpra.
  }
  if(sum(dados[,4]<0)>=1){ # Controle de fluxo que interrompe a execução da
função se pelo menos um dos valores de tamanho populacional é menor ou igual
a 0.
    stop("Erro: Existe pelo menos um valor de tamanho populacional que não é
maior do que 0") # Interrope a função e emite a mensagem de erro entre
aspas, caso a condição anterior do controle de fluxo se cumpra.
  }
  ##### CÁLCULOS BÁSICOS E COMPARTILHADOS ENTRE OS MODOS DE FUNÇÃO:
  if(input=="freq" | input=="raw"){ # Controle de fluxo que determina a
execução desta porção do código quando o argumento 'input' é igual a "freq"
e a "raw". Na prática o código é executado deste ponto em diante para ambos
os argumentos válidos do argumento 'input'.
    ### Transformando frequências genotípicas em valores absolutos de
contagem de indivíduos:
    pops=length(dados[,1]) # Cria um objeto contendo um valor inteiro,
representando o número de linhas (e portanto de populações) no objeto de
entrada 'dados'.
    ngo=data.frame(rep(NA, pops), rep(NA, pops), rep(NA, pops)) # Cria um
data.frame contendo apenas valores NA, para ser preenchido com os valores de
número de indivíduos observados para cada genótipo (nas colunas 1, 2 e 3),
para cada população (nas linhas).
    for (l in 1:pops){ # Cria um loop com um contador 'l', que irá ciclar de
1 até o valor armazenado no objeto 'pops'.
      for (c in 1:3){ # Cria um loop com um contador 'c' que irá ciclar nas
3 primeiras colunas do objeto 'dados'.
        ngo[l,c]=dados[l,c]*dados[l,4] # Cria um data.frame contendo um
```

```
valor inteiro, representando valores de contagem de indivíduos para cada
genótipo, para cada população. Através da multiplicação dos valores de
frequência pelo tamanho populacional de cada população.
      }
    }
    ### Cálculo de p e q (frequências dos alelos 1 e 2):
    af=data.frame(p=rep(NA, pops), q=rep(NA, pops)) # Cria um data.frame
contendo apenas valores NA, para ser preenchido com os valores de frequência
de cada alelo p e q (nas colunas 1 e 2), para cada população (nas linhas).
    for (l in 1:pops){ # Cria um loop com um contador 'l', que irá ciclar de
1 até o valor armazenado no objeto 'pops'.
      p=(ngo[l,1]*2+ngo[l,2])/(dados[l,4]*2) # Calcula o valor da frequência
do alelo 1 (p) para a população da linha "l" e atribui ao objeto 'p'. O
calculo é o seguinte: (2 * nº de indivíduos homozigotos para o alelo 1 da
população "l" + nº de indivíduos heterozigotos da população "l") / (2 * o
tamanho populacional da população "l").
      q=1-p # Calcula o valor da frequência do alelo 2 (q) para a população
da linha "l" e atribui ao objeto 'q'.
      af[l,1]=p # Adiciona 'p' a posição localizada na linha "l" e coluna 1
do objeto 'af'.
      af[l,2]=q # Adiciona 'q' a posição localizada na linha "l" e coluna 2
do objeto 'af'.
    }
    ### Cálculo das frequências genotípicas esperadas:
    nge=data.frame(rep(NA, pops), rep(NA, pops), rep(NA, pops)) # Cria um
data.frame contendo apenas valores NA, para ser preenchido com os valores de
número de indivíduos esperados sob Equilíbrio de Hardy-Weinberg para cada
genótipo (nas colunas 1, 2 e 3), para cada população (nas linhas).
    fge=nge=data.frame(rep(NA, pops), rep(NA, pops), rep(NA, pops)) # Cria
um data.frame contendo apenas valores NA, para ser preenchido com os valores
de frequências genotípicas esperadas sob Equilíbrio de Hardy-Weinberg (nas
colunas 1, 2 e 3), para cada população (nas linhas).
    for (l in 1:pops){ # Cria um loop com um contador 'l', que irá ciclar de
1 até o valor armazenado no objeto 'pops'.
      D=(af[l,1]^2)*dados[l,4] # Calcula o valor de indivíduos homozigotos
esperados para o alelo 1, para a população da linha "l" e atribui ao objeto
'D'. O calculo é o seguinte: p² * n, onde "p" é a frequência do alelo 1 e
"n" é o tamanho populacional da população "l".
      H=(2*af[l,1]*af[l,2])*dados[l,4] # Calcula o valor de indivíduos
heterozigotos esperados, para a população da linha "l" e atribui ao objeto
'H'. O calculo é o seguinte: 2*p*q * n, onde "p" é a frequência do alelo 1,
"q" é a frequência do alelo 2 e "n" é o tamanho populacional na população
"l".
      R=(af[l,2]^2)*dados[l,4] # Calcula o valor de indivíduos homozigotos
esperados para o alelo 2, para a população da linha "l" e atribui ao objeto
'R'. O calculo é o seguinte: q² * n, onde "q" é a frequência do alelo 2 e
"n" é o tamanho populacional da população "l".
      nge[l,1]=D # Adiciona 'D' a posição localizada na linha "l" e coluna 1
do objeto 'nge'.
      nge[l,2]=H # Adiciona 'H' a posição localizada na linha "l" e coluna 2
do objeto 'nge'.
```

```
      nge[l,3]=R # Adiciona 'R' a posição localizada na linha "l" e coluna 3
do objeto 'nge'.
      fge[l,1]=D/dados[l,4] # Calcula a frequência genotípica esperada de
homozigotos do alelo 1, para a população da linha "l" e adiciona o valor a
posição localizada na linha "l" e coluna 1 do objeto 'fge'.
      fge[l,2]=H/dados[l,4] # Calcula a frequência genotípica esperada de
heterozigotos, para a população da linha "l" e adiciona o valor a posição
localizada na linha "l" e coluna 3 do objeto 'fge'.
      fge[l,3]=R/dados[l,4] # Calcula a frequência genotípica esperada de
homozigotos do alelo 2, para a população da linha "l" e adiciona o valor a
posição localizada na linha "l" e coluna 3 do objeto 'fge'.
    }
    ##### PRODUZINDO GRÁFICOS:
    ### Produzindo gráficos das frequências genotípicas e alélicas
observadas, e genotípicas esperadas sob Equilíbrio de Hardy-Weinberg, para
cada população:
    if(graphics=="on"){ # Controle de fluxo que determina a execução desta
porção do código quando o argumento 'graphics' é igual a "on".
      transparent <- rgb(0, 0, 0, alpha=0) # Cria um objeto contendo uma
string de caracteres que designa uma "cor transparente".
      for (i in 1:pops){ # Cria um loop com um contador 'i', que irá ciclar
de 1 até o valor armazenado no objeto 'pops'.
        barplot(as.matrix(dados[i,c(1:3)]), col = "grey", ylim = c(0,1),
main = paste(rownames(dados[i,])), space = 0, las = 1, cex.axis = 1) # Cria
um gráfico de barras, para cada população, contendo os valores de
frequências genotípicas observadas.
        barplot(as.matrix(fge[i,c(1:3)]), col = transparent, ylim = c(0,1),
main = paste(rownames(dados[i,])), add = T,axes = F, axisnames = F, space =
0, border = "red") # Adiciona aos gráficos criados no passo anterior, as
frequências genotípicas esperadas de cada população.
        legend("topright", legend = c("Observado", "Esperado"), col =
c("grey", "red"), pch = 15, cex = 1) # Cria uma legenda para as frequências
genotípicas observadas ("Observado") e esperadas ("Esperado").
        barplot(as.matrix(af[i,c(1:2)]), col = "grey", ylim = c(0,1), main =
paste(rownames(dados[i,])), space = 0, las = 1, cex.axis = 1) # Cria um
gráfico de barras, para cada população, com as frequências alélicas (p e q).
      }
    }
    if(graphics=="off"){ # Controle de fluxo que determina a execução desta
porção do código quando o argumento 'graphics' é igual a "off".
      warning("Os gráficos estão desabilitados") # Emite uma mensagem
avisando que a construção de gráficos está desativada.
    }
    ##### MODOS DE OPERAÇÃO DA FUNÇÃO E OUTPUTS:
    if(modo=="ehw"){ # MODO EHW: Controle de fluxo que determina a execução
desta porção do código quando o argumento 'modo' é igual a "ehw".
      ### Calculando o x² e o p-valor:
      output=data.frame("Obs-11"=rep(NA, pops),"Obs-12"=rep(NA,
pops),"Obs-22"=rep(NA, pops),
                        "Esp-11"=rep(NA, pops),"Esp-12"=rep(NA,
```

```
pops),"Esp-22"=rep(NA, pops),
                        "x^2"=rep(NA, pops), "p-valor"=rep(NA, pops),
"EHW"=rep(NA, pops)) # Cria um data.frame contendo apenas valores NA, para
ser preenchido com o número de indivíduos observados, esperados, qui-
quadrado calculado, p-valor e um valor lógico dizendo se a população está ou
não em EHW.
      rownames(output)=rownames(dados) # Atribui ao objeto 'output' os nomes
das linhas do objeto 'dados', os quais são os nomes das populações
originais.
      x2=c() # Cria um vetor vazio, para conter temporariamente os valores
de qui-quadrado, a ser usado no loop a seguir.
      for (l in 1:pops){ # Cria um loop com um contador 'l', que irá ciclar
de 1 até o valor armazenado no objeto 'pops'.
        for (c in 1:3){ # Cria um loop com um contador 'c', que irá ciclar
de 1 a 3.
          output[l,c]=ngo[l,c] # Atribui o número de indivíduos observados
para cada genótipo ("c"), em cada população ("l"), contidos no objeto 'ngo'
às posições "l" x "c" do objeto 'output'.
          output[l,c+3]=nge[l,c] # Atribui o número de indivíduos esperados
sob EHW para cada genótipo ("c"), em cada população ("l"), contidos no
objeto 'ngo' às posições "l" x ("c"+ 3) do objeto 'output'.
          x2[c]=((output[l,c]-output[l,c+3])^2)/output[l,c+3] # Faz o
cálculo de x² para cada população e armazena no vetor 'x2' em cada posição
"c". O cálculo de x² de aderência é feito por ((Nº Observado - Nº
esperado)^2/Nº esperado), para cada genótipo, em cada uma das populações.
          output[l,7]=sum(x2) # O cálculo de x² de aderência total para cada
população é dado pelo somatório dos valores de x² encontrados para cada
genótipo.
          output[l,8]=pchisq(output[l,7], df=1) # O p-valor para cada
população é estimado a partir do valor de x² calculado também para cada
população, com grau de liberdade igual a 1 (argumento 'df').
          output[l,9]=(output[l,8]<1-alfa) # Por fim é realizado um teste
lógico, comparando o p-valor estimado para cada população com o valor de
significância do teste (argumento 'alfa'), definido pelo usuário. Por padrão
o valor de alfa é 0.05.
        }
      }
      return(output) # Retorna o objeto 'output' como saída da função.
    }
    if(modo=="fit"){ # MODO FIT: Controle de fluxo que determina a execução
desta porção do código quando o argumento 'modo' é igual a "fit".
      ### Cálculo FIS, FST e FIT:
      HI=sum(ngo[,2])/sum(dados[,4]) # Cálculo da média observada de
heterozigotos através da seguinte fórmula: número de heterozigotos
observados em cada população / número total de indivíduos.
      HS=sum(nge[,2])/sum(dados[,4]) # Cálculo da média esperada sob EHW de
heterozigotos através da seguinte fórmula: número de heterozigotos esperados
em cada população / número total de indivíduos.
      pmean=sum(af[,1])/pops # Cálculo do valor médio de frequência do alelo
1 entre todas as populações (p médio). O cálculo é feito através da soma dos
valores de frequência do alelo 1 (p) em cada população, divida pelo número
```

```
total de populações.
      qmean=1-pmean # Cálculo do valor médio de frequência do alelo 2 entre
todas as populações (q médio). O valor é dado pelo complemento do p médio,
ou seja, 1 - p médio.
      HT=2*pmean*qmean # Heterozigosidade esperada para a população como um
todo, em outras palavras, quando se considera que todas as populações são na
verdade são apenas uma única população maior.
      FIS=(HS-HI)/HS # Calcula o FIS como a redução relativa entre a
heterosigosidade esperada média e heterosigosidade observada média em cada
população.
      FST=(HT-HS)/HT # Calcula o FST como a redução relativa entre a
heterosigosidade esperada na população como um todo e considerando a média
das heterosigosidades esperadas para cada subpopulação, considerando-se a
estruturação em subpopulações.
      FIT=(HT-HI)/HT # Calcula o FIT como a redução relativa entre a
heterosigosidade esperada na população como um todo e a heterosigosidade
observada média em cada população.
      output=data.frame("FIS"=FIS, "FST"=FST, "FIT"=FIT) # Cria um dataframe
contendo os valores de FIS, FST, FIT em cada coluna e com os respectivos
nomes das colunas.
      rownames(output)="VALORES" # Renomeia a linha contendo os valores com
a string de caracteres "VALORES".
      return(output) # Retorna o objeto 'output' como saída da função.
    }
  }
}
```

Arquivos da função

Função genpop()

Help da função genpop()

Arquivo dos exemplos do help